



How to Install WordPress Locally

KINSTA



For more information on how to grow your website and your business,
scan the QR code above or go to:

kinsta.com/ebooks

Published by **KINSTA**

How to Install WordPress Locally

Contents

CHAPTER 1

How to Install WordPress Locally on Mac, Windows, and Linux With DevKinsta	7
Download and Install DevKinsta	8
Create a Local WordPress Site With DevKinsta	9
New WordPress Site	10
Import From Kinsta	11
Custom Site	11
Navigate DevKinsta's Site Info Screen	13
Manage Multiple Sites in DevKinsta	15
Push Your Site to MyKinsta	16
Use Adminer to Manage Your Database	19
Inspect Outgoing Email From WordPress	24

CHAPTER 2

How to Install WordPress Locally on Mac and Windows With DesktopServer	28
Download DesktopServer to Your Computer	29
Launch the DesktopServer Installer	30
Enable Plugins and Start Apache and MySQL Services	32
Create a New Development Site	33
Install and Configure Your WordPress Site	35
Additional Tips for Using DesktopServer to Install WordPress Locally	37

CHAPTER 3

How to Install WordPress Locally on Windows Using WAMP	40
Download and Install WAMP on Your Computer	40
Run the Wampserver.exe File to Start the Installation	41
Create a New MySQL Database	42
Install WordPress and Extract the Files	44
Visit Your Local WordPress Site in Your Web Browser	45

CHAPTER 4

How to Install WordPress Locally on Mac Using MAMP	47
Download and Install MAMP on Your Computer	47
Launch MAMP and Start Your Servers	48
Create Your Database and Update Your User Information	50
Install WordPress and Visit Your Site from Localhost	52

CHAPTER 5

How to Install WordPress Locally on Mac, Windows, and Linux Using XAMPP	55
Download and Install XAMPP on Your Computer	55
Choose the Components You Want to Install	56
Launch the XAMPP Control Panel and Test Your Server	57
Download WordPress and Create a Database	58
Install WordPress Locally by Visiting Your Site in Your Browser	59
Conclusion	60

Local development is a great way to build and debug offline in a safe environment that doesn't use any hosting resources.

There are several reasons local development might be advantageous to you. Whether you want to test out new features, experiment with development projects, or build a WordPress site before pushing it live, a local WordPress install can help you do that. If you're traveling and don't have access to Wi-Fi, you could use a local install to continue working. A local install may also be faster when it comes to manipulating files.

You can install WordPress locally in just a few simple steps.

To do so, you'll need to get a local AMP stack set up on your machine. In the case of WordPress, AMP stands for Apache, MySQL, and PHP. These are the softwares needed to mimic what a [managed WordPress host](#) would be running for you on its web server.

In this ebook, you'll learn how to install WordPress locally on Windows, Mac, and Linux using DevKinsta, DesktopServer, XAMPP, WAMP, or MAMP.

How to Install WordPress Locally on Mac, Windows, and Linux With DevKinsta

[DevKinsta](#) is Kinsta's very own [local development tool for WordPress](#). You can use DevKinsta to create local WordPress sites with a single click. Its features include advanced database and email management tools as well as a full integration with [MyKinsta](#).

DevKinsta is completely free, even if you're not a Kinsta customer. It's available for macOS, Windows, and Ubuntu Linux. DevKinsta is also currently used by more than 10,000 developers, web designers, and freelancers.

Being able to pull sites hosted with Kinsta, push them to staging, and create a local environment has made my life soooooo much easier. Reviewing plugins/themes and endlessly putting them on domains is now a thing of the past. Even if you're not hosting with Kinsta, I'd recommend the tool.



Ben Townsend

Here are a few key features:

- One-click WordPress site creation with multisite and [WP-CLI](#) support
- A modern stack powered by [Nginx](#), [MySQL](#), and the latest versions of [PHP](#)
- MyKinsta integration so you can import Kinsta-hosted sites, and push changes to Kinsta
- Database management with [Adminer](#)
- [SMTP server](#) and email capture tool for inspecting outgoing emails

Download and Install DevKinsta

To get started, [download the latest version of DevKinsta](#). The tool is available in 10 different languages, all of which can be found by toggling the language option in the footer.

- To install DevKinsta on macOS, open the **.dmg** file, and drag the DevKinsta app into your **Applications** folder. Double-click on DevKinsta in your **Applications** folder.
- To install DevKinsta on Windows, double-click on the DevKinsta executable (**.exe**) and step through the installation wizard.
- To install DevKinsta on Ubuntu, download the **.deb** package and install it. This can be done on the command-line, or with your preferred package installer.

When you start DevKinsta for the first time, [Docker Desktop](#) will be installed as a dependency. DevKinsta uses Docker Desktop for creating containerized WordPress environments.

During the DevKinsta installation process, you may see a pop-up message that says, “Docker Desktop needs privileged access.” If you see that message, click **Okay** and provide the password for your user account so Docker Desktop can be installed correctly.

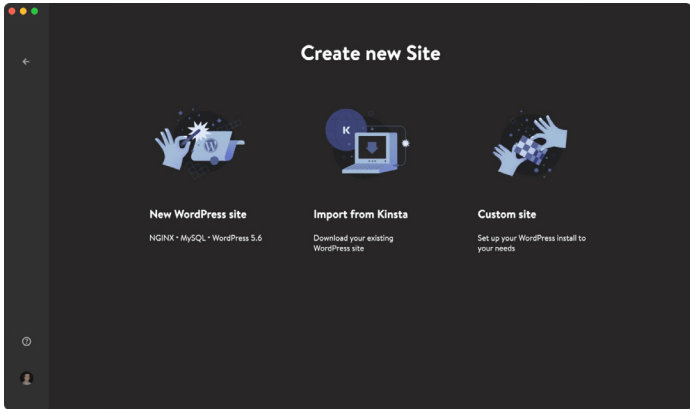
After you provide the password for installation, DevKinsta will install Docker Desktop along with some Docker images. The installation can take some time depending on the speed of your internet connection, so feel free to step away from the computer for a while.

Create a Local WordPress Site With DevKinsta

DevKinsta supports three methods for creating local WordPress sites.

Note: During the site creation process, you may be prompted to provide your user password or confirm permissions depending on your version of macOS or Windows. On some operating systems, DevKinsta requires extended permissions to write your site’s files to disk.

1. **New WordPress Site:** Create a local site with the default hosting stack consisting of Nginx, MySQL, PHP 7.4, and the latest version of WordPress.
2. **Import from Kinsta:** Clone a site hosted on Kinsta to your local computer with just a few clicks. After you’re finished with your work, you can even push changes back to a Kinsta staging environment.
3. **Custom Site:** Create a local site with a customized [hosting stack](#). With this option you can [choose your preferred PHP version](#), specify your database name, and enable [WordPress multisite](#).

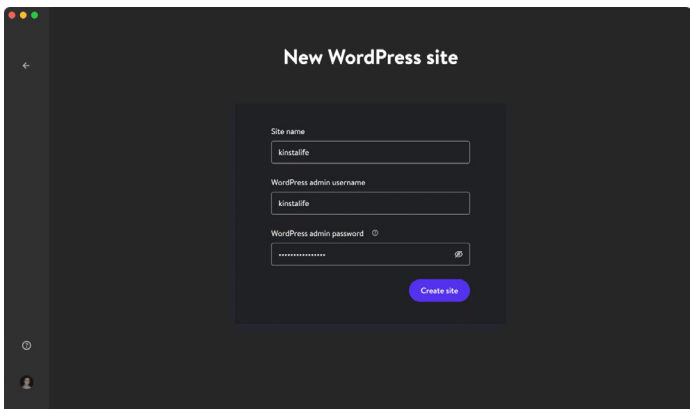


DevKinsta has three methods for creating local WordPress sites

Here's a closer look at each site creation option.

New WordPress Site

To get started, select the [New WordPress Site option](#). For this site creation method, all you have to do is specify a site name, WordPress admin username, and WordPress admin password. Once you've filled in these three fields, click **Create Site**.



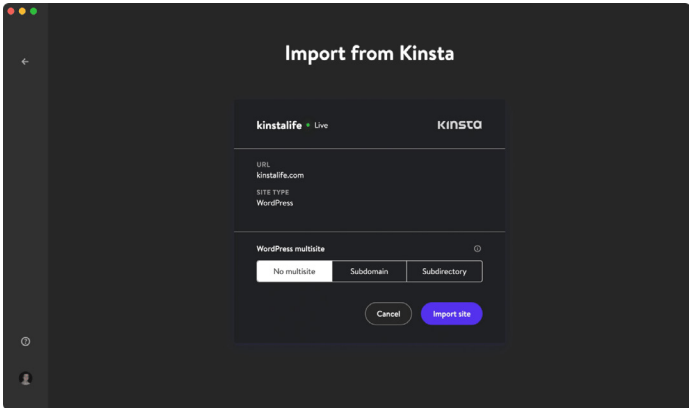
Create a new WordPress site in DevKinsta

Import From Kinsta

The second option is to import a site environment that's already hosted on Kinsta. To do this, [click Import from Kinsta](#) and provide your MyKinsta login details.

Once logged in, select the Kinsta environment you'd like to clone to your local computer. DevKinsta supports both live and staging environments on Kinsta, so be sure to choose the correct one.

After clicking on an environment, specify whether the site is a multisite installation and click **Import Site** to start cloning your site.



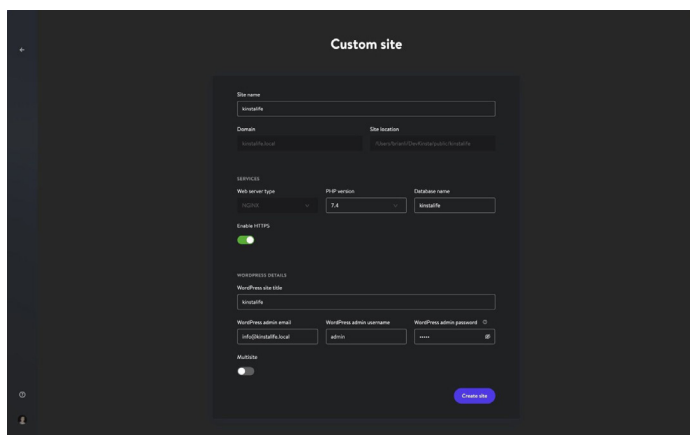
Clone your live site with the "Import from Kinsta" feature

Custom Site

With the third option, you can configure specific settings for your local WordPress installation.

Here are the settings you can tweak with **Custom Site**:

- Site Name
- PHP Version (PHP 7.2, 7.3, 7.4, 8.0)
- Database Name
- Enable HTTPS
- WordPress Site Title
- WordPress Admin Email
- WordPress Admin Username
- WordPress Admin Password
- WordPress Multisite Mode



Customize a local WordPress installation with DevKinsta

After configuring your desired settings, click **Create Site** to start the site creation process.

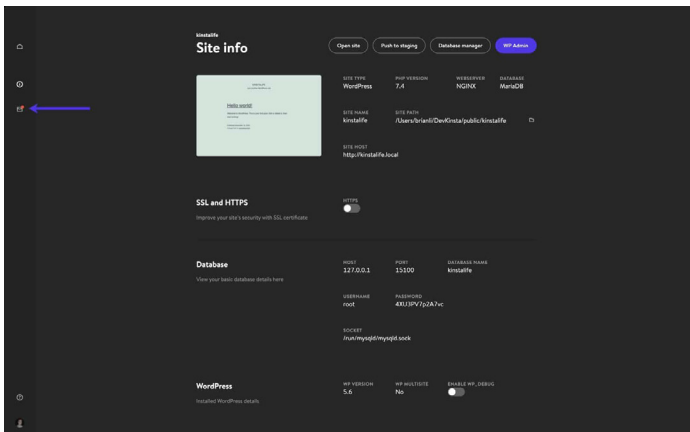
Navigate DevKinsta's Site Info Screen

After creating a site, you'll see the **Site Info** screen. Each site created in DevKinsta has its own **Site Info** page. Think of this screen as the mission control dashboard for your local WordPress site.

Within **Site Info**, you can find useful information like site identity details, PHP version, WordPress version, **SSL mode**, database credentials, and site host name.

The **Site Info** screen also has convenient buttons for opening your local site in a web browser, pushing a site to a **Kinsta staging environment**, launching **Adminer** for database management, and accessing the WordPress admin dashboard of your local WordPress install.

Let's walk through the key aspects of each section of the **Site Info** screen.



The "Site Info" screen in DevKinsta

The top of the **Site Info** screen has general information about your WordPress site. For developers, **Site Path** and **Site Host** are especially useful.

Site Path refers to the location of the WordPress install on the local filesystem. You can click on the folder icon to go straight to the folder and start editing themes, plugins, and more.

The **Site Host** is a custom **.local** domain name (e.g. <https://kinstalife.local>) that you can use to access a local WordPress site in a web browser.

The **SSL and HTTPS** section contains an HTTPS toggle, which automatically generates an [SSL certificate](#) for your local WordPress site, and allows you to access the site over HTTPS.

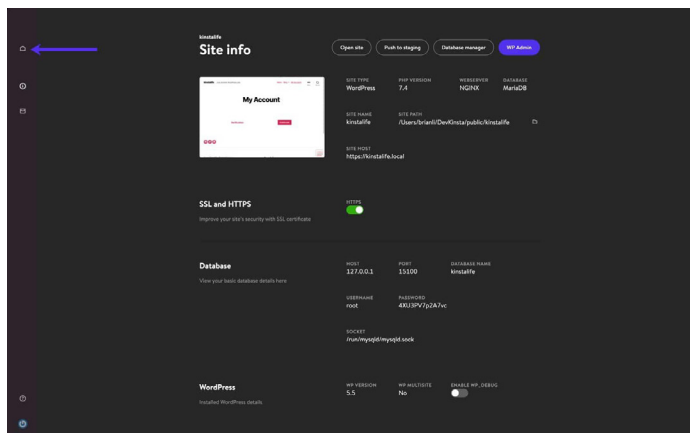
The **Database** section displays the database settings for your local WordPress site. This is useful if you want to access your [WordPress database](#) via the MySQL command line tool or a third-party database management tool.

Lastly, the **WordPress** section displays your WordPress Core version and multisite mode status. And there's even a toggle for enabling [WP_DEBUG mode](#) to troubleshoot your WordPress site.

Manage Multiple Sites in DevKinsta

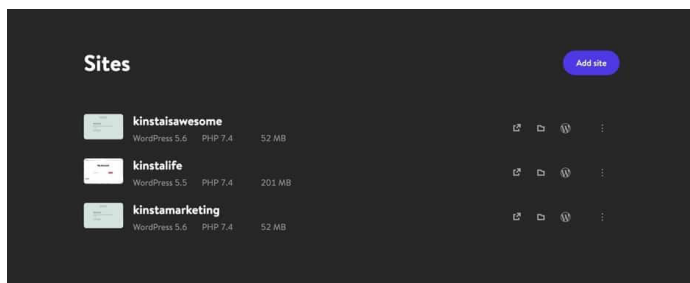
For agencies and developers, or anyone working on multiple projects at once, DevKinsta can be used to deploy and manage multiple local WordPress sites. Every local WordPress site managed by DevKinsta runs in its own containerized environment. This means every site has its own customizable PHP version, WordPress version, email inbox, and more.

To view your DevKinsta site list, click on the site's icon in the left-hand sidebar.



Deploy multiple WordPress local environments with DevKinsta

On this screen, you can see a list of all your local WordPress sites. To add another, click the **Add Site** button.



Manage multiple local WordPress sites with DevKinsta

Push Your Site to MyKinsta

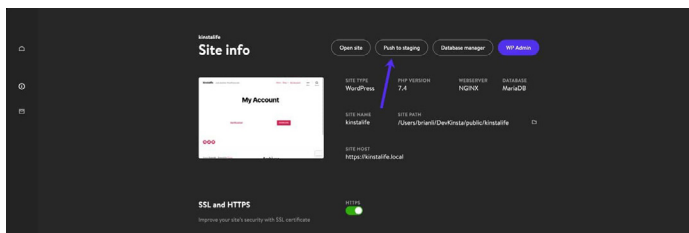
If your WordPress sites are hosted on Kinsta, DevKinsta makes it easy to [push changes online to a Kinsta staging environment](#).

DevKinsta offers an easy way to create and manage your site for all WordPress users. If you're a Kinsta customer, you can have the entire DevKinsta experience. You can push your local site to your dedicated Kinsta staging environment, as well as pull your live sites to your local environment. This means you have a full path open when it comes to going from local to live and back again. This is invaluable, especially with tight turnarounds or clients wanting to give feedback on your work.



Tom Rankin

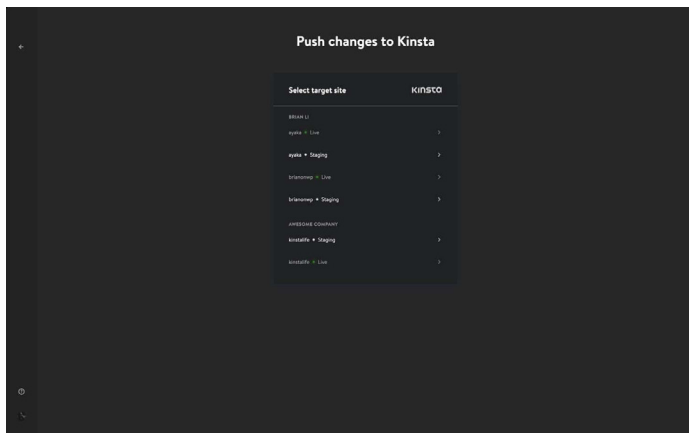
To push a local site to Kinsta, click the **Push to Staging** button on your **Site Info** page.



Push your local WordPress site to a Kinsta staging environment

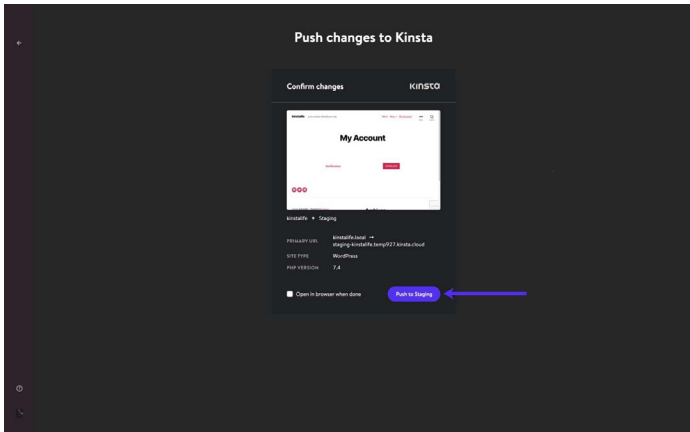
You may be prompted to enter your MyKinsta credentials.

Then, select a target site to push to. Keep in mind that this process will overwrite the contents of the current staging environment if one exists.



Choose a staging environment to push changes to

Finally, click **Push to Staging** to confirm the action.



Confirm the "Push to Staging" action

Once you've pushed your local WordPress site to Kinsta, you can view the site via the staging environment URL. When ready, you can then [push staging to live in MyKinsta](#).

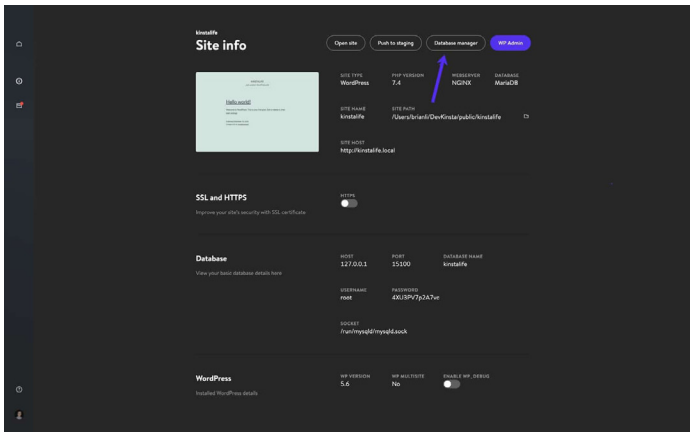
See how easy it is to push staging to live
in our free MyKinsta demo



Use Adminer to Manage Your Database

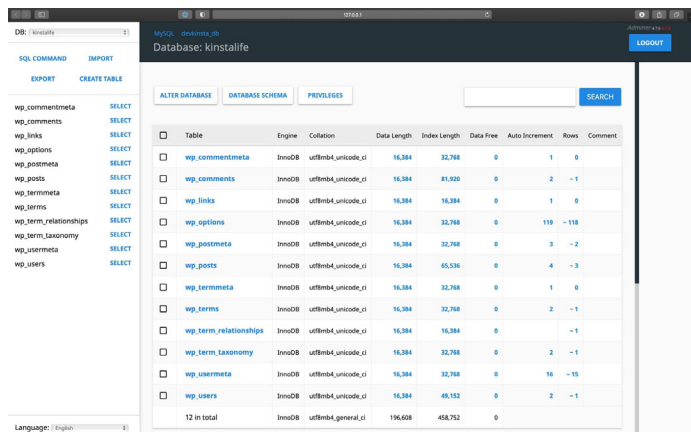
DevKinsta ships with a lightweight database management tool called [Adminer](#). Like [phpMyAdmin](#), which we use for sites hosted on Kinsta, Adminer provides you with a web interface to edit database tables, run database queries, import and export backups, and more.

To launch Adminer, click the [Database Manager](#) button at the top of the **Site Info** page. Adminer will then open in your default web browser.



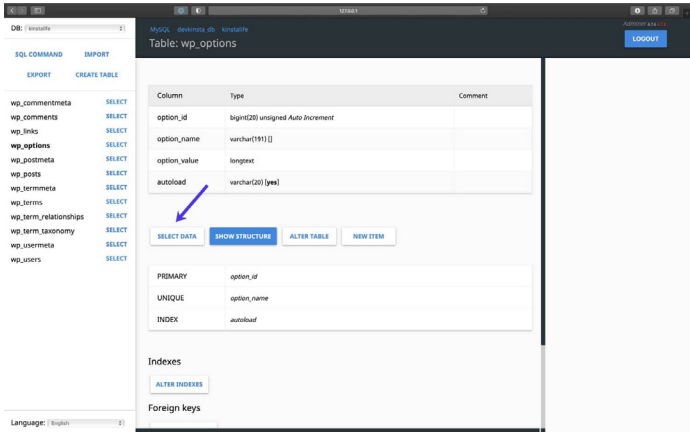
Click “Database Manager” to access Adminer in DevKinsta

After launching Adminer, you'll see your WordPress database's tables. The screenshot below shows the database of our "kinstalife" test site. Under the **Table** column, you can see the default WordPress tables like **wp_comments**, **wp_posts**, etc.:



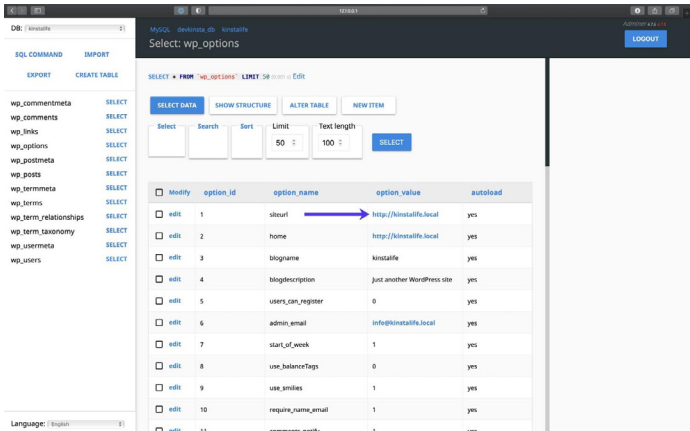
WordPress database in Adminer

To edit a database entry, click on the desired table. For example, if we want to edit the home and site URL of our WordPress site, we can click on the **wp_options** table.



Click "Select Data" to edit your WordPress database tables

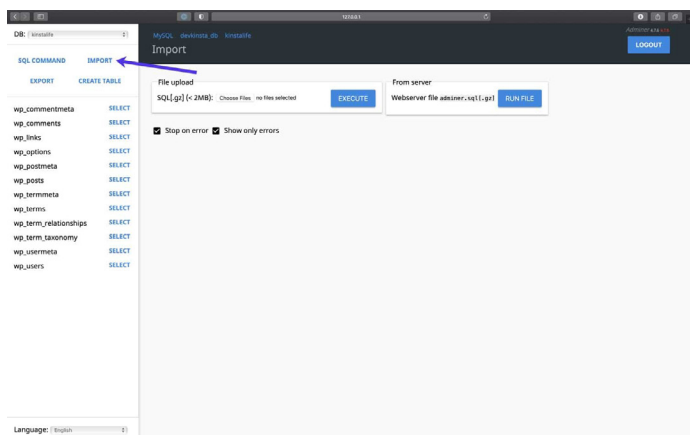
On this page, we can edit the **option_value** for **siteurl** to update the site URL of our WordPress site, and the same can be done for the home URL as well.



Edit a WordPress database "option_value" with Adminer

Adminer also supports database import and export as well. This is useful for working with database backup files like the ones we include with [downloadable backups](#).

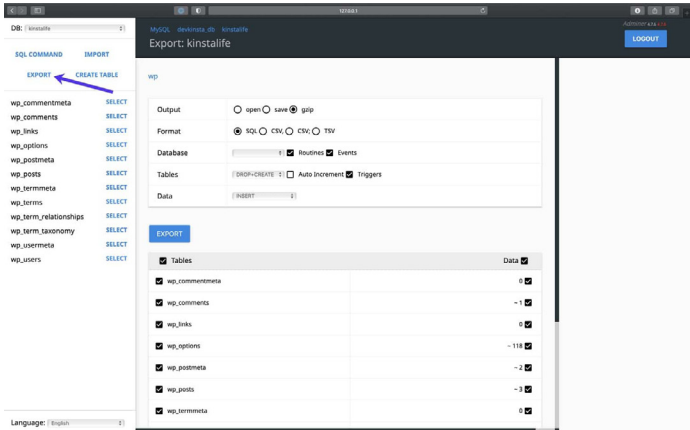
To import a database file, click **Import** in the upper left corner of Adminer. Click on **Choose Files** to select a database backup, and click **Execute** to start the import process. Adminer supports both raw **.sql** files as well compressed **.sql.gz** files.



Import a database backup with Adminer

To export a complete database backup, click **Export** in the upper left corner of Adminer. Select **gzip** for the output format, **SQL** for the database format, and leave the other settings as-is. Click **Export** to start the backup process.

Adminer will then export your WordPress database to a compressed **.sql.gz** file.

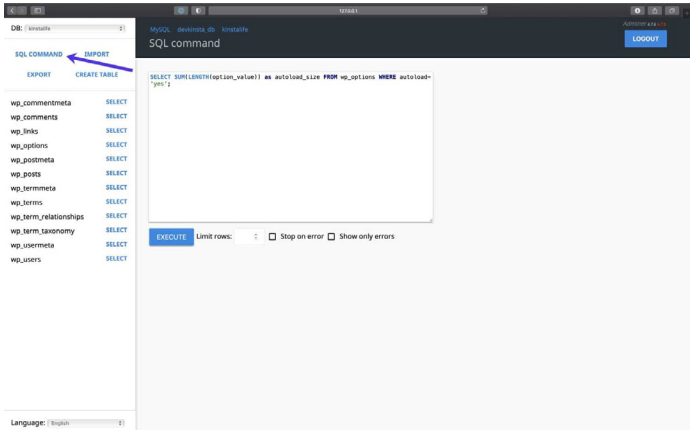


Export a database backup from Adminer

Finally, Adminer supports SQL command execution, which means you can run database queries on your WordPress database. For example, if you're trying to find the amount of [autoloaded data](#) in your database, you can run the below SQL command in Adminer.

```
SELECT SUM(LENGTH(option_value)) as autoload_size  
FROM wp_options WHERE autoload='yes';
```

To run a database query, click **SQL Command** in the upper left corner of Adminer. Specify a database query, and click **Execute** to run the command.



Query your database with SQL commands in Adminer

With DevKinsta's Adminer integration, you have advanced control over your WordPress database.

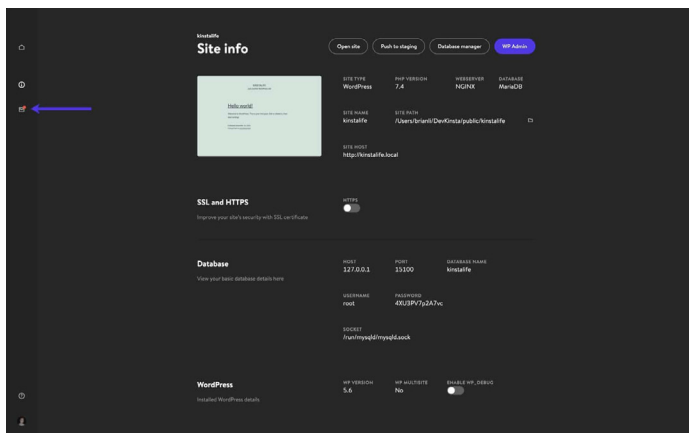
Whether you need to edit database tables, import or export backups, or run complex SQL commands, DevKinsta has got you covered!

Inspect Outgoing Email From WordPress

DevKinsta includes a [built-in SMTP server and email capture tool](#). This means your local WordPress sites can send outgoing emails like a live production site. But in this case, sent emails will be captured and stored in DevKinsta's email inbox.

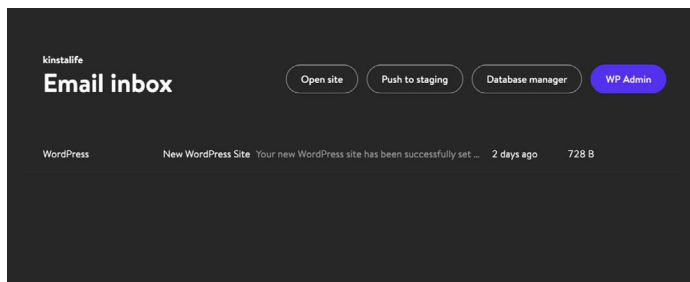
This gives you the best of both worlds: you can use DevKinsta to test outgoing email functionality for [marketing automation workflows](#), [WooCommerce](#) order confirmations, and more without spamming the email inboxes of your visitors and customers.

To access DevKinsta's email inbox, click on the mail icon in the left-hand sidebar.



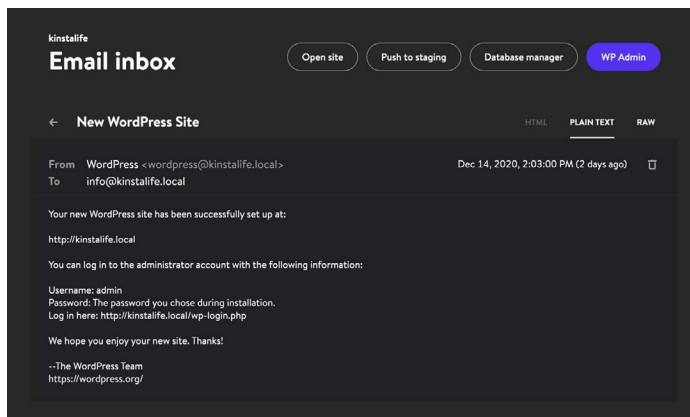
DevKinsta includes a built-in SMTP server and email capture tool

In the email inbox, you'll see a list of outgoing emails that were captured. In the screenshot below, you can see an outgoing email from our “kinstalife” test site:



An outgoing email in DevKinsta's email inbox

To inspect an outgoing email, simply click on it. In each email, DevKinsta lets you inspect the “from” address, “to” address, body content, time of delivery, and more.



DevKinsta email inbox display modes

You can also choose to display the email in HTML, Plain Text, or Raw mode. The HTML mode is useful for testing out HTML email templates, while the Raw mode lets you inspect email headers like MIME-Version and X-Mailer directly.

To learn more about DevKinsta, be sure to check out our [official community forum](#), read [DevKinsta documentation](#), and watch our [helpful DevKinsta tutorial videos on YouTube](#).

And if you haven't already, [download DevKinsta](#).

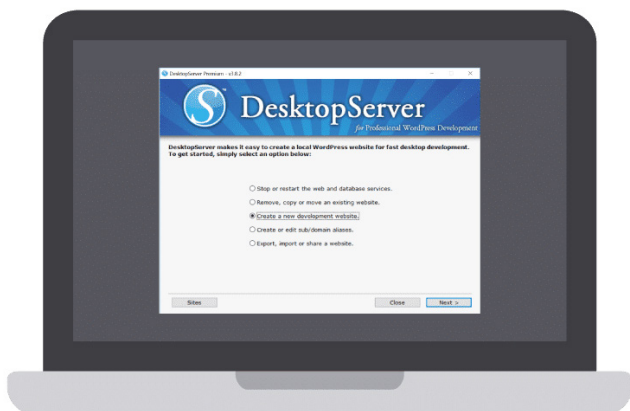
DevKinsta is an excellent free local WordPress development tool. I'm a big fan of Kinsta — both the hosting and the people who run it. This tool has made my local/prod sync super easy.



Ahmad Awais

How to Install WordPress Locally on Mac and Windows With DesktopServer

DesktopServer is a great WordPress product by ServerPress, which can make installing WordPress locally on Mac and Windows a complete breeze.



The DesktopServer screen

You can launch a new development install in a matter of seconds with the simple click of a button. This tool also fully **supports multisite** and **WP-CLI**.

ServerPress has both a free version and a premium version, the latter costing \$99.95 per year. The premium version includes a few advanced capabilities, such as:

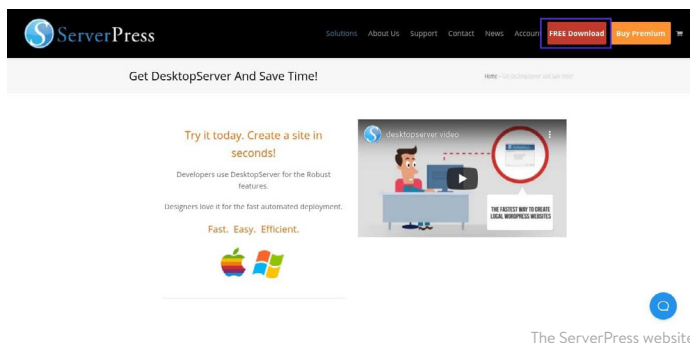
- Multisite support
- Importing and exporting third-party backups
- Deploying directly to your live site
- Bypassing any login plugins

You can choose which version fits you best depending on your needs. If you just need to do some quick testing, the free version works well.

Download DesktopServer to Your Computer

To install WordPress locally, [download DesktopServer](#) from their website. There is both a Windows version and a Mac version. For this example, we'll be using the Windows version.

On the website, click the **Free Download** button in the top right-hand corner (or Buy Premium if that's what you prefer):



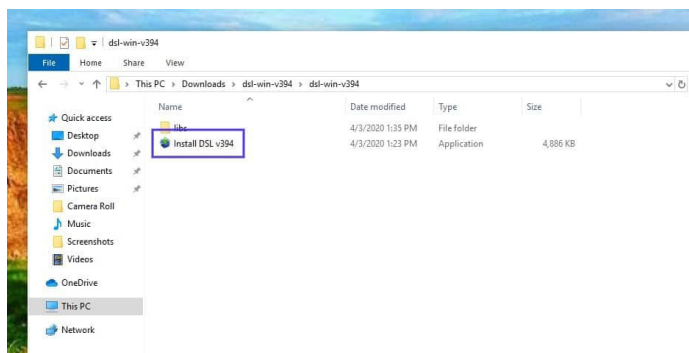
The ServerPress website

This will automatically begin downloading the **.zip** file to your computer.

Launch the DesktopServer Installer

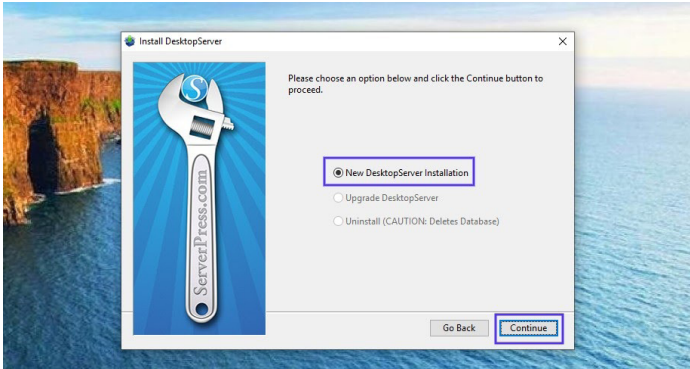
Once the file has finished downloading, the next step is to launch the DesktopServer installer. Before you do that, unzip the file you just downloaded. This may take a few minutes to complete.

Once that's finished, click on **Install DSL**.



The DesktopServer installer application

When you first launch the program, you'll be prompted to restart with administrator privileges. Select **Continue**. You'll then be prompted to accept the terms of services and choose an option for your installation.



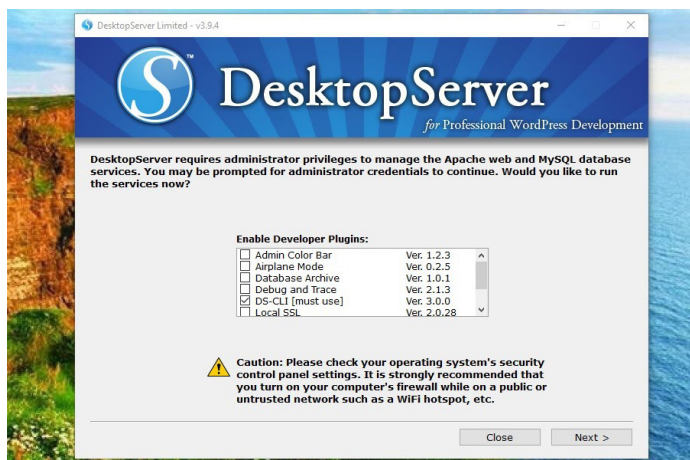
The DesktopServer installation window

Leave **New Desktop Installation** selected, then click on **Continue**. The installation process will begin, which can take a bit of time.

When it's finished, a pop-up will appear letting you know it's complete. It also tells you where in your computer directory you can find the application. When you're done, click on **Finish**.

Enable Plugins and Start Apache and MySQL Services

Once the installation is complete, you can enable a multitude of different developer plugins.



The DesktopServer developer plugins screen

Here's a quick rundown of the developer plugins you can use when you install WordPress locally – we highly recommend the bypass login and DS-CLI plugins:

- **Airplane mode:** Control loading of external files when developing locally
- **Bypass login:** Allows developer bypass of login credentials, via quick selection of any of the first 100 usernames in a combobox
- **Clean Import:** Resets `.htaccess`, clears cache from third-party hosts
- **Debug and Trace:** Forces `WP_DEBUG = true`, and enables cross-platform/language trace statements in PHP and JavaScript

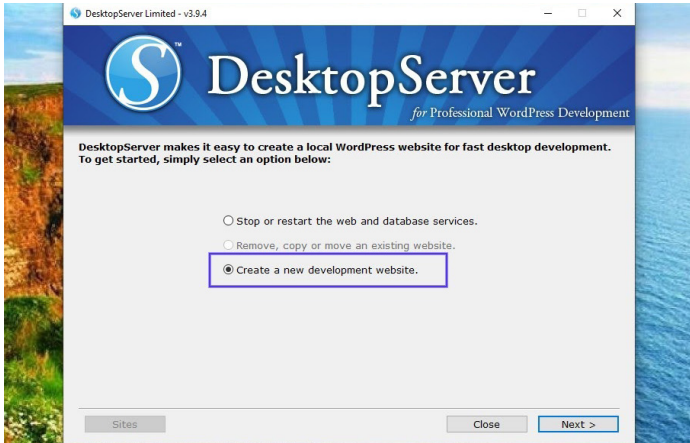
- **Dreamweaver Support:** Enables automatic Dreamweaver project file creation, and a WYSIWYG mode when working on template files and style.css
- **DS-CLI:** This is an enhanced, cross-platform, command-line interface for professional developers. It lets you easily [use CLI](#), Composer, [Git](#), and PHPUnit. NodeJS and NPM are included to allow installation of GRUNT, Gulp, and other Node dependencies
- **DS-Deploy:** Used to move a site from a local DesktopServer install to a live server
- **InnoDB Autoconvert:** [Convert a site's tables to InnoDB](#) on Create, Copy, Move, and Import operations
- **Local Admin Color Bar:** Changes the Admin bar color
- **Mailbox Viewer:** Provides quick developer offline viewing of mail delivery services

Keep in mind that some of these options are only available with the premium version.

When you're done, select **Next**. You'll then be asked if you want to start web and database services, so hit **Next** again.

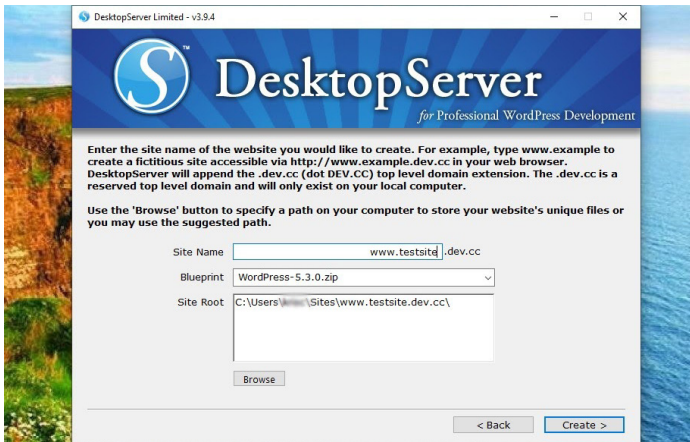
Create a New Development Site

When you're done enabling plugins and starting the web and database services, the next prompt will be to select "Create a new development website."



The option to create a new development website in DesktopServer

This is where the program will install WordPress for you. You'll have to pick your site's name, which will also be its local address. We are calling ours "testsite," so our development URL will be "testsite.dev" on our local machine:



The option to create a new development website in DesktopServer

DesktopServer enables you to actually create different blueprints, making it almost like a pre-built template. In our case, however, we simply want a fresh install.

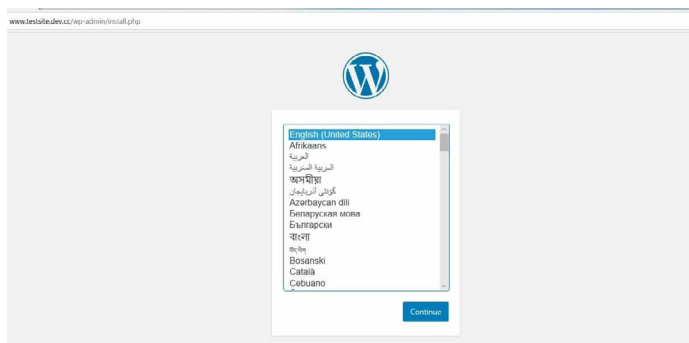
DesktopServer always has the latest version of WordPress as the default blueprint. This means you don't have to worry about manually downloading it from the repository and unzipping it.

By default, the site's root is located in your "My Documents" folder. If you're happy with this, you can leave it as is. However, for ease of organization, we changed ours to a folder we created in the root of our C: drive called "wordpress."

When you're ready, click on **Create**. You'll then see the URL of your local WordPress install. Click on that to finish the installation.

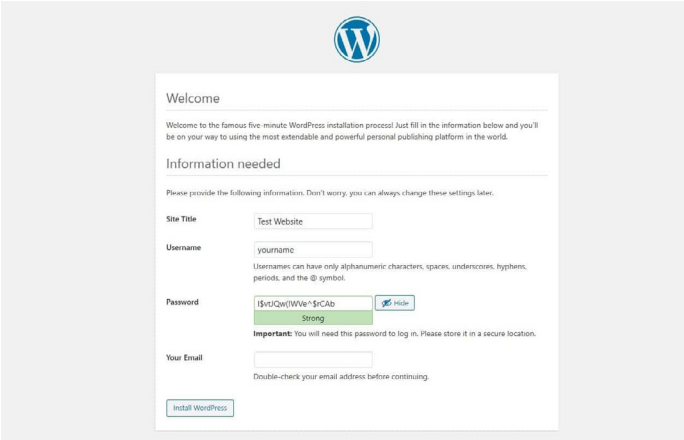
Install and Configure Your WordPress Site

When you click on the link we just mentioned, your local WordPress site will open in a browser tab.



A new WordPress installation setup page

After you choose your language, the next step is to give your site a title and choose a username, a strong password, and your email address (if you plan to make the site live later, avoid using “admin” as a username, which you can read more about in our [WordPress security guide](#)).



Welcome

Welcome to the famous five minute WordPress installation process! Just fill in the information below and you'll be on your way to using the most extendable and powerful personal publishing platform in the world.

Information needed

Please provide the following information. Don't worry, you can always change these settings later.

Site Title

Username

Usersnames can have only alphanumeric characters, spaces, underscores, hyphens, periods, and the @ symbol.

Password
Strong

Important: You will need this password to log in. Please store it in a secure location.

Your Email

Double-check your email address before continuing.

The welcome page of a new WordPress site

When you're done, select **Install WordPress**.

That's it! You just installed WordPress locally and your site is up and running. You can now browse to your local install and test away.

In our case, we'll go to “testsite.dev” in our browser's address bar. Since we selected the bypass login plugin feature during setup, there's a [dropdown menu](#) where we can select our admin and be automatically logged in. Obviously you won't use this on a production site, but it's super handy for a dev environment.

Additional Tips for Using DesktopServer to Install WordPress Locally

Because of how Windows handles its file permissions, you may or may not see a message [upon logging in](#) about [WordPress failing to update](#).

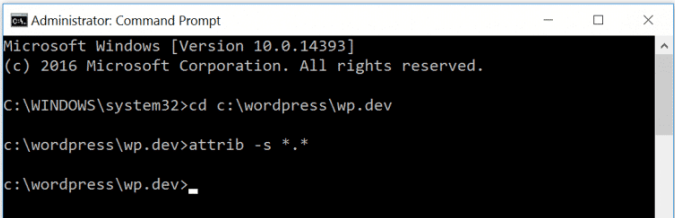


An automated WordPress update has failed to complete - [please attempt the update again now](#).

A WordPress update failed message

To fix, this simply open up the command prompt as an administrator and run the following command within your WordPress directory folder:

```
attrib -s *.*
```



```
Administrator: Command Prompt
Microsoft Windows [Version 10.0.14393]
(c) 2016 Microsoft Corporation. All rights reserved.

C:\WINDOWS\system32>cd c:\wordpress\wp.dev

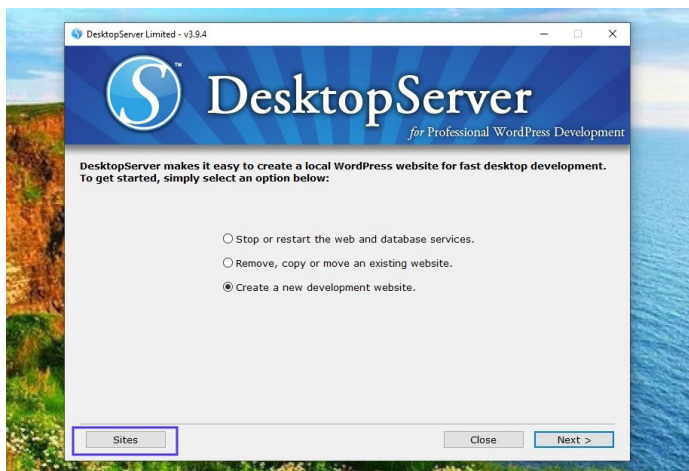
c:\wordpress\wp.dev>attrib -s *.*

c:\wordpress\wp.dev>
```

LAMP permissions for WordPress

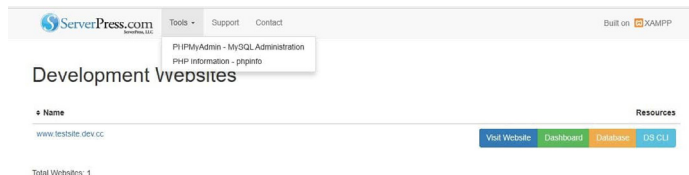
If you need more guidance or instruction on how to do this, you can [learn more about the details on ServerPress](#).

To create additional WordPress sites or edit them, simply launch the **DesktopServer.exe** file again. You can stop and restart services, create new sites, edit them, export and import them, etc. To [access phpMyAdmin](#), you can click on the **Sites** button on the bottom left:



LAMP permissions for WordPress

Alternatively, you can enter “localhost” into your browser’s address bar. This will bring up the administrator interface on the [localhost](#).



The administrator interface of the DesktopServer localhost

There, you can get the links to all of your WordPress sites, as well as dashboard links and phpMyAdmin links.

Another awesome feature is the ability to launch [WP-CLI](#) (or DS-CLI) with a single click. If you selected the **DS-CLI** option during the setup process above, there will be a link for it within your dashboard. Click on it to start firing up WP-CLI commands.

DesktopServer also includes an export feature, which you can use to export your WordPress site directly to a live site or **.zip** file. Note that you will need the premium version for this.

How to Install WordPress Locally on Windows Using WAMP

[WampServer](#), also known as WAMP, is a software that bundles [Apache web server](#), PHP, and MySQL specifically for Windows devices.

Download and Install WAMP on Your Computer

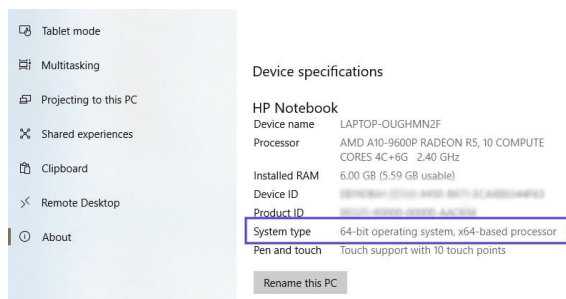
To download and install the WAMP software, [visit the WampServer website](#) and select **Start Using WampServer**.



The WampServer website

This will automatically bring you to the downloads section of the site, where you'll have two versions to choose from: WampServer 32-bit and WampServer 64-bit. Select the one that's recommended for your operating system.

If you're unsure whether your OS is 32-bit or 64-bit, navigate to **Settings > About**.



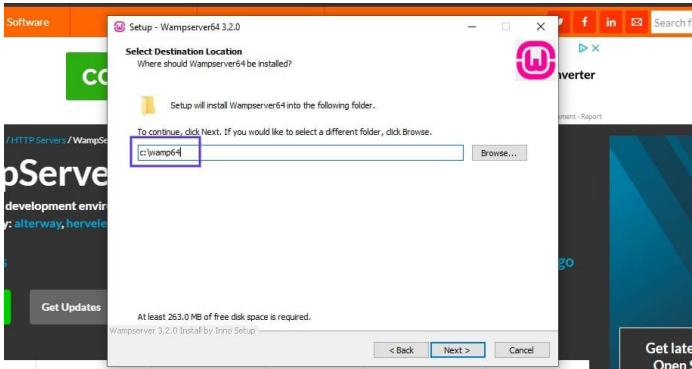
The device specifications page on Windows

Under the **Device specifications** section, you'll be able to find your OS type.

Run the Wampserver.exe File to Start the Installation

After you download the software, click on the wampserver.exe file to run the installer. This may take a minute or two.

Also, make note of where this file downloads to, as you'll need to revisit it later.



The device specifications page on Windows

You'll be prompted with a series of instructions on the screen to complete the installation process.

During this process, you'll be asked to define a web browser. You can always change this option to a browser you prefer by navigating to the **Program Files** of your computer.

Create a New MySQL Database

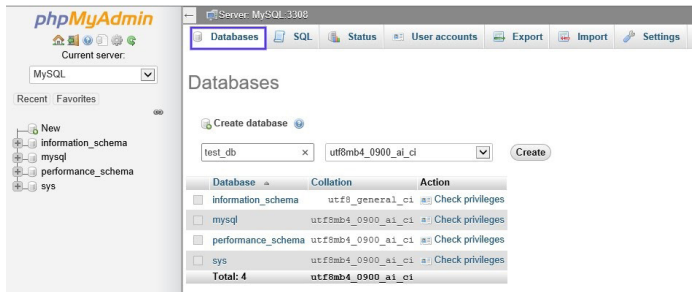
The next step is to set up a blank [MySQL database](#). After you launch WAMP, there will be a green icon in the bottom right-hand corner of your screen.

Click on the icon, followed by **phpMyAdmin**. This will automatically bring you to the login screen in your browser.



The phpMyAdmin login page

In the username field, input “root,” leave the password field empty, and then select the **Go** button. Next, click on **Databases**.



The Databases page of phpMyAdmin

Under the **Create Databases** section, you’ll need to name your new database. Next, click on **Create**. You now have your **database** set up.

Install WordPress and Extract the Files

Once you're done creating your database, the next step is to install WordPress locally. To do this, visit WordPress.org and click on **Get WordPress**, followed by **Download WordPress**.



The download page on WordPress.org

This will download a **.zip** file to your computer. To extract the files, click on the folder and select **Extract All**.

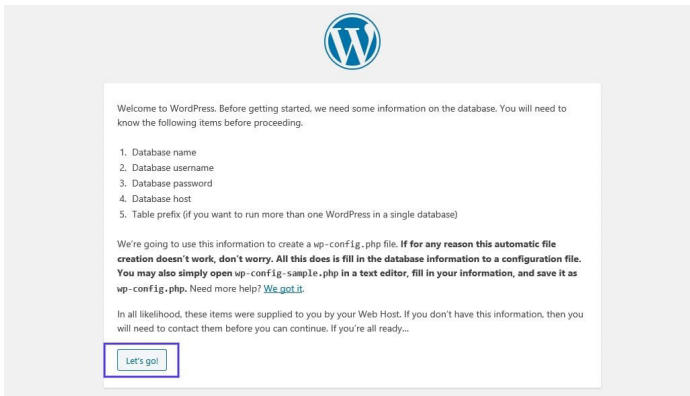
When that's done, right-click on the folder and select **Copy**. Navigate back to the folder on your computer where you downloaded WAMP and paste the "wordpress" folder into that directory.

At this point, you can also rename the "wordpress" folder if you wish. The name of the folder will be the URL for your local WordPress install. For this tutorial, we'll rename ours to "mytestsite."

Visit Your Local WordPress Site in Your Web Browser

Open your web browser and type into the search bar “http://localhost/mytestsite/.” Replace “mytestsite” with whatever you named your “wordpress” folder.

The software will then present a series of prompts for you to set up your [WordPress installation](#). You’ll select a language and review the database information (the same series of steps we discussed in the previous section). When you’re done, click on **Let’s go!**



The database details page of a new WordPress installation

On the next screen, you’ll enter your database information. The name will be what you called your database, the username is “root,” and you can leave the password field empty.

Next, click on the **Run the installation** button. Then, name your site and create a username and password.

When you're finished, select **Install WordPress**. When the software is done installing, it will show you a Success! message.

You can then click on **Log In**. This will bring you to the admin login page for your WordPress site.

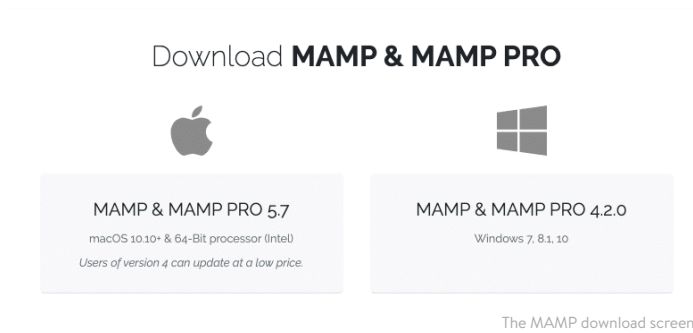
That's it! You now have a local testing environment installed.

How to Install WordPress Locally on Mac Using MAMP

If you're looking for local server software to use on a Mac computer, you might consider MAMP. MAMP is short for Macintosh, Apache, MySQL, and PHP. It's incredibly polished and easy to use.

Download and Install MAMP on Your Computer

As with the two previous local development options, the first step is to download and install MAMP on your computer. You can do this [from the official MAMP website](#).

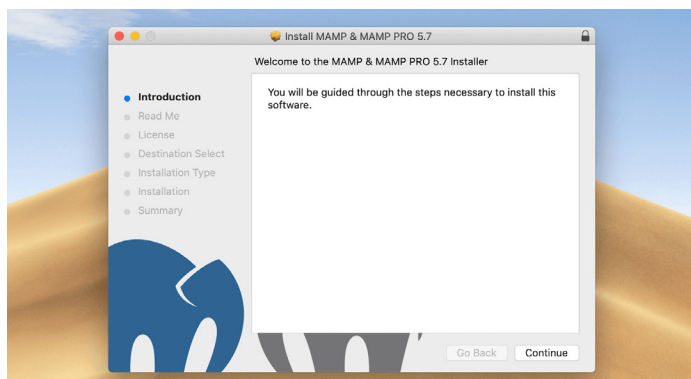


The MAMP download screen

Note that while you can download and use MAMP for free, there are also [premium plans](#) available.

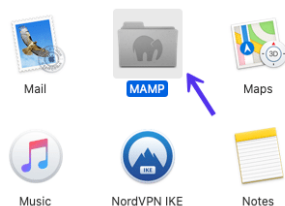
Launch MAMP and Start Your Servers

Once it's finished downloading, click on the **mamp.pkg** file. An installation window will pop up. Select the **Continue** button to follow the series of prompts.



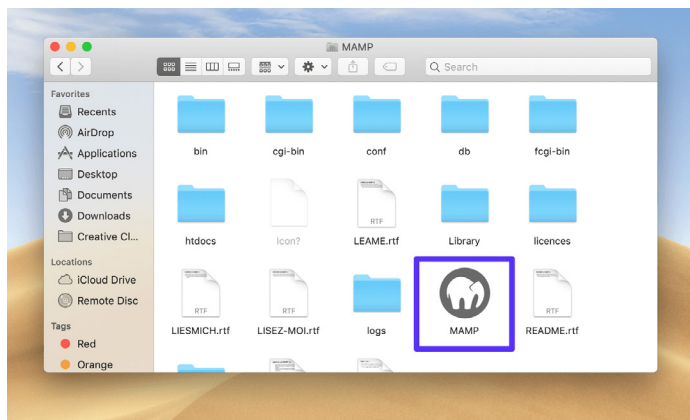
The MAMP installation window

Next, navigate to **Go > Applications** on your computer and click on the MAMP folder.



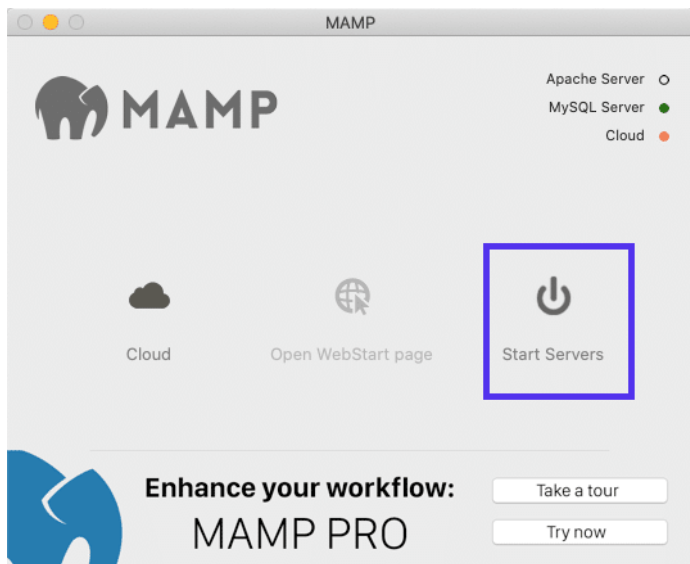
The MAMP application folder

Inside that folder, click on the MAMP elephant icon.



The MAMP application icon

This will open a new window. Hit **Start Servers**.

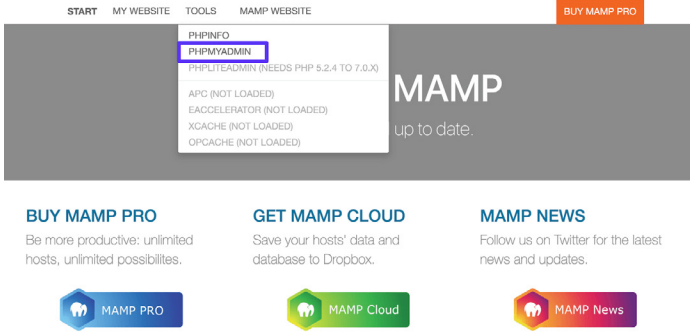


The MAMP "Start Servers" option

Once the Apache and MySQL servers launch, MAMP will automatically open the WebStart page in your browser.

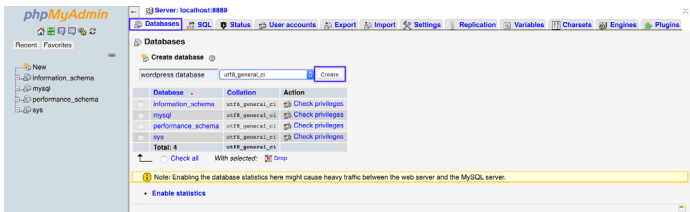
Create Your Database and Update Your User Information

Now it's time to create a new database. On the WebStart page, select **Tools > phpMyAdmin**.



Opening phpMyAdmin via the MAMP WebStart page

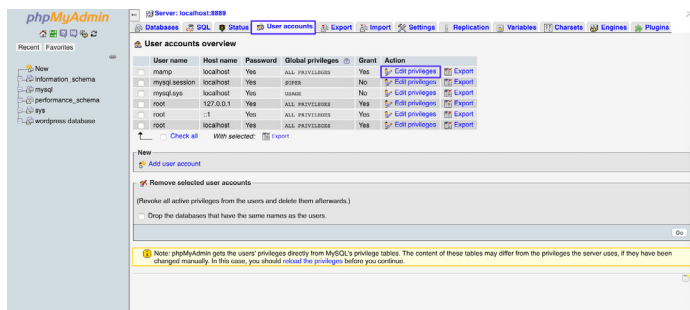
Once phpMyAdmin opens, click on the **Databases** tab. Name your database, and then select **Create**.



Creating a new database for your local MAMP site

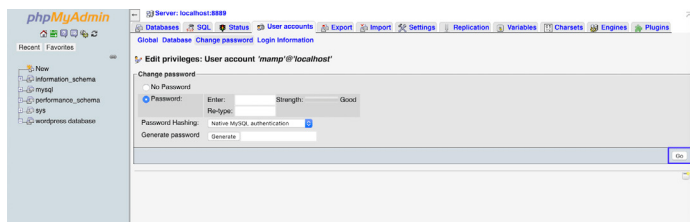
Next, you'll need to update the MySQL database user credentials for the default account MAMP creates for you, as you'll need them to complete the WordPress installation process. Navigate back to the phpMyAdmin home screen and click on the **User Accounts** tab.

Then, click on **Edit privileges** for the account with the username **mamp**.



Editing the default MAMP phpMyAdmin user account

Select the **Change password** tab, enter your preferred password, and hit **Go**.



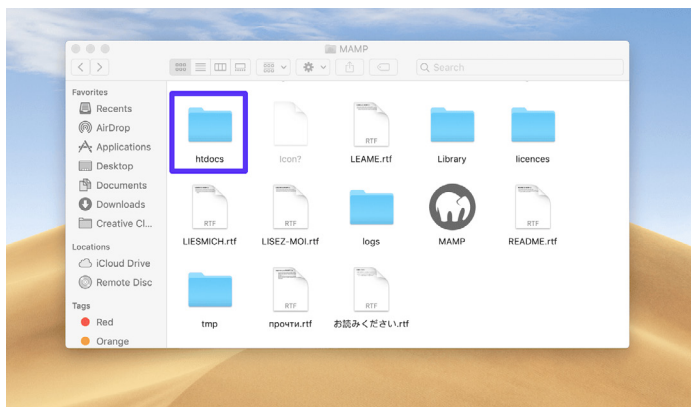
Changing the default MAMP phpMyAdmin account password

You can then close phpMyAdmin.

Install WordPress and Visit Your Site from Localhost

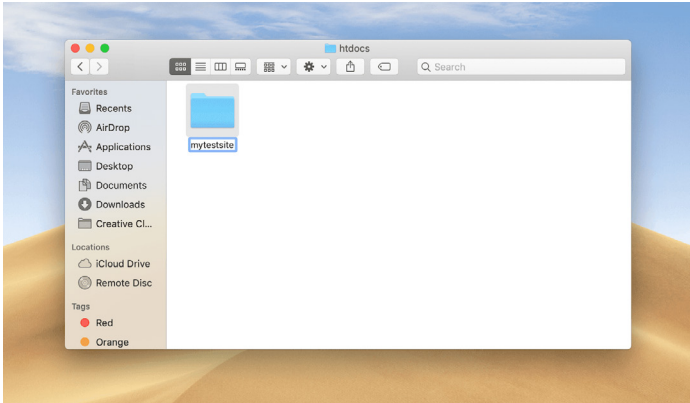
Now, visit the [WordPress.org website](https://wordpress.org) and download the latest version of WordPress. Unzip the “wordpress” folder once it’s finished downloading. Right-click on the folder and select **Copy**.

Navigate back to **Go > Applications > MAMP** on your computer and open the **htdocs** folder.



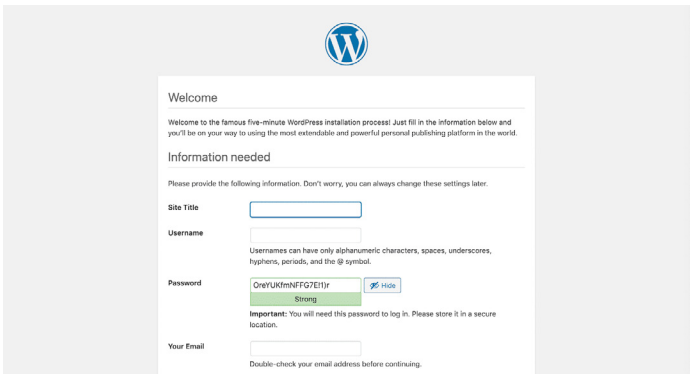
The htdocs folder in the MAMP application

Inside that folder, paste your WordPress folder that you just copied. We suggest renaming it to “mytestsite” or something similar.



Renaming the MAMP local WordPress installation

Then, browse to “http://localhost/8888/mytestsite” in a new browser tab. Here you’ll be prompted to input your database credentials, as well as name your site.



Completing the WordPress installation wizard

Once you complete the WordPress installation prompts, you're all done! If you need more instructions on this step, you can refer to the previous section in this ebook.

How to Install WordPress Locally on Mac, Windows, and Linux Using XAMPP

XAMPP is another popular PHP development environment you can use to install WordPress locally. You can use it for Windows, macOS, or Linux. Here, we'll walk you through how to do it for Windows, although the process is largely the same for Mac users.

Download and Install XAMPP on Your Computer

Visit the [Apache Friends](#) website, and next to the green **Download** button, select **XAMPP for Windows** (or whichever OS you're using).



XAMPP Apache + MariaDB + PHP + Perl

What is XAMPP?


XAMPP is the most popular PHP development environment


XAMPP is a completely free, easy to install Apache distribution containing MariaDB, PHP, and Perl. The XAMPP open source package has been set up to be incredibly easy to install and to use.




Download

Click here for other versions

 XAMPP for Windows
7.4.8 (PHP 7.4.8)

 XAMPP for Linux
7.4.8 (PHP 7.4.8)

 XAMPP for OS X
7.4.8 (PHP 7.4.8)

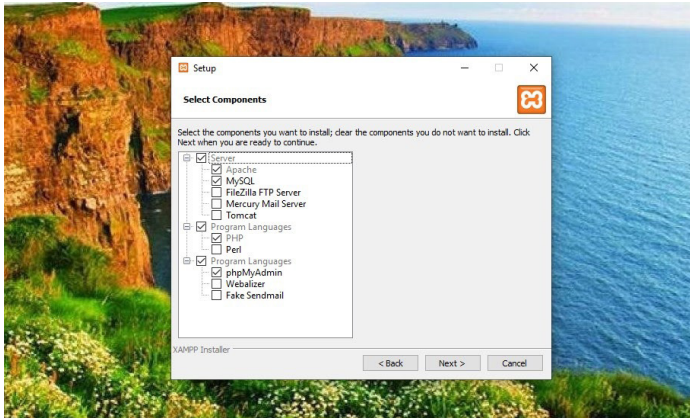
The Apache Friends website

The software will automatically download to your computer. When it's finished, click on the **.exe** file to launch the installer.

Note that for macOS, this will be a **.dmg** file. Once you open it, click on the XAMPP icon and drag it to your **Applications** folder.

Choose the Components You Want to Install

After running the installer, you'll be asked to choose the components to install. The most important ones to select are **Apache**, **MySQL**, **PHP**, and **phpMyAdmin**.



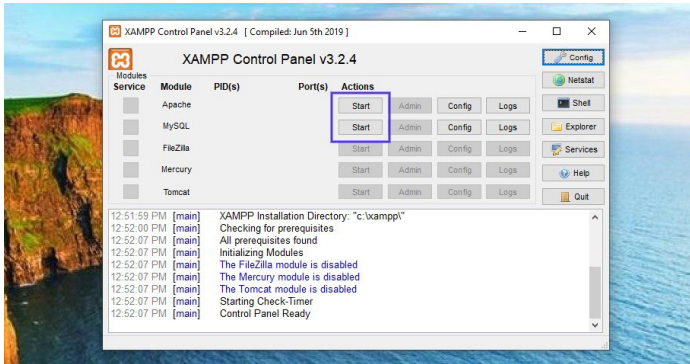
The XAMPP set up components screen

You can uncheck the other components, as they aren't necessary. When you're done, click on the **Next** button and select which folder you want to install XAMPP on.

Click on the **Next** button again, ignore the Bitnami prompt, and select **Next** once again.

Launch the XAMPP Control Panel and Test Your Server

On the final screen, choose to launch the XAMPP Control Panel. Once the XAMPP Control Panel has opened, you can click the **Start** buttons next to both **Apache** and **MySQL**.



The XAMPP Control Panel

After you launch them, the status for each should turn to green. Now it's time to test your server. You can do this by entering "http://localhost/" into your web browser.

If it works, you have successfully added XAMPP to your computer.

Download WordPress and Create a Database

The next step is to install WordPress on your computer. You can do this by going to WordPress.org and clicking on **Get WordPress**.

When the package is finished downloading, extract the files, and then copy the folder. Next, navigate to the XAMPP folder on your computer, and locate and open the **htdocs** folder.

Next, create a new folder within the **htdocs** folder. You can name it something along the lines of "mytestsite." Within that folder, paste the WordPress files.

Now it's time to create your database.

Navigate back to your XAMPP control panel and select **Admin** next to **MySQL**. This will launch phpMyAdmin.

Click on **Databases**, then name your database and select **Create** (you can refer to previous sections if you need more guidance).

You can name your database whatever you want. However, we suggest keeping it simple and memorable, such as “test_db.”

Install WordPress Locally by Visiting Your Site in Your Browser

To complete the process, you can visit “http://localhost/mytestsite” in your browser. Remember to replace “mytestsite” with whatever you named your WordPress folder.

You'll be prompted to select a language, name your site, and fill in your database details. Then you can log into your WordPress site and start using your local environment!

Conclusion

Whether you're headed to a location with limited Wi-Fi or want to quickly edit and manipulate files, a local WordPress installation can make your life easier.

While many local development tools are available in the market, we'd highly recommend [DevKinsta](#). It's free forever, seamlessly integrates with MyKinsta, and is available in 10 languages.

[Download DevKinsta](#)



Need a blazing-fast, secure, and developer-friendly hosting for your client sites? Kinsta is built with [WordPress developers in mind](#) and provides plenty of tools and a powerful dashboard.

Some of our features include:

- Nginx
- PHP 8.0, 7.4, and 7.3 (check out our [PHP benchmarks](#))
- [One-click staging](#)
- MariaDB
- [Full Page Caching](#)
- [SSH Access](#)
- [Object Caching](#) (with [Redis](#) add-on)
- SSL + SPDY Support
- A+ on the Qualys SSL Labs test
- [Free migration and hack fixes](#)
- Free [Application Performance Monitoring \(APM\) Tool](#)
- [Manual backup points](#)

- Composer Support
- Free Premium DNS via Route 53
- [DevKinsta Local Development Tool](#)
- HTTP/3 Support
- Latest software versions

Even more details can be found on our [Advanced Features](#) page.

[Check out our plans](#)



The background is a gradient of teal and dark blue. It features several semi-transparent, overlapping geometric shapes, including rectangles and squares, in various shades of teal and dark blue, creating a layered, abstract effect.

KINSTA